Amendments to the Claims:

Please amend Claims 1, 6, 8, 10, 17, 19; cancel Claim 7, as shown below.

1. (Currently Amended): A method for implementing a two-phase commit protocol, comprising:

associating a plurality of resources in a transaction, wherein the plurality of resources are applied in the transaction using a prepare phase and a commit phase, wherein the prepare phase comprises a plurality of prepare operations, each of which is associated with one of the plurality of resources, and wherein the commit phase comprises a plurality of commit operations, each of which is also associated with one of the plurality of resources;

dispatching a [[first]] second prepare operation from a first server thread to a second server thread, wherein the [[first]] second prepare operation is associated with a [[first]] second resource and [[a]] the prepare phase, and the second prepare operation is executed by sending a preparation instruction from the second server thread to the second resource;

processing a [[second]] first prepare operation by the first server thread in parallel to the [[first]] second prepare operation being processed by the second server thread, wherein the [[second]] first prepare operation is associated with a [[second]] first resource and the prepare phase, and the first prepare operation is executed by sending another prepare instruction from the first server thread to the first resource;

determining that the prepare phase is complete when both the first prepare operation and the second prepare operation are completed;

after determining that the prepare phase is complete, dispatching a [[first]] second commit operation from the first server thread to a third server thread, wherein the [[first]] second commit operation is associated with the [[first]] second resource and [[a]] the commit phase, and the second commit operation is executed by sending a commit instruction from the third server thread to the second resource;

processing a [[second]] first commit operation by the first server thread in parallel to the [[first]] second commit operation being processed by the third server thread, wherein the [[second]] first commit operation is associated with the [[second]] first resource and the commit phase, and the first commit operation is executed by sending another commit instruction from the first server thread to the first resource; and

~~after~~ determining that the commit phase is complete when both the first commit operation and the second commit operation are completed~~, writing results of the commit phase to a~~

~~transaction log~~.

2. (Previously Presented): The method of claim 1 further comprising:
selecting an idle server thread to process the first prepare operation.

3. (Original): The method of claim 2, wherein selecting includes:
determining available server threads in a server.

4. (Previously Presented): The method of claim 3 wherein a thread pool manager determines the available server threads in the server.

5. (Previously Presented): The method of claim 1 further comprising:
reporting results of the prepare phase to a log.

6. (Currently Amended): A method for processing [[N]] a two-phase commit protocol operations, comprising:

associating a plurality of resources in a transaction, wherein the plurality of resources are applied in the transaction using a prepare phase and a commit phase, wherein the prepare phase comprises a plurality of prepare operations, each of which is associated with one of the plurality of resources, and wherein the commit phase comprises a plurality of commit operations, each of which is also associated with one of the plurality of resources;

processing [[N]] the plurality of prepare operations in a first server thread, ~~wherein each of the N prepare operations are associated with a prepare phase,~~ wherein the processing [[for]] of each prepare operation of [[N-1]] the plurality of the prepare operations ~~includes~~ comprises:

dispatching the prepare operation to ~~a second~~ another server thread if ~~a second~~ the another thread is ~~determined to be~~ available; and

processing the prepare operation in the first server thread if no ~~second~~ other server thread is ~~determined to be~~ available;

~~processing a remaining prepare operation in the first server thread~~;

determining that the prepare phase is complete, when every one of the plurality of prepare operations completes;

after determining that the prepare phase is complete, processing [[N]] the plurality of commit operations in a first server thread, ~~wherein each of the N commit operations are associated with a commit phase,~~ wherein the processing [[for]] of each commit operation of [[N-1]] the plurality of the

commit operations ~~includes~~ comprises:

dispatching the commit operation to ~~a second~~ another server thread if ~~a second~~ another server thread is determined to be available;

processing the commit operation in the first server thread if no ~~second~~ other server thread is ~~determined to be~~ available;

~~processing a remaining commit operation in the first server thread;~~ and

~~after~~ determining that the commit phase is complete, when every one of the plurality of commit operations completes ~~results of the commit phase are written to a transaction log~~.

7. (Canceled).

8. (Currently Amended): The method of claim [[7]] 6 wherein a thread pool manager determines the available server threads in the server.

9. (Canceled).

10. (Previously Presented): The method of claim 6 further comprising:

reporting results of the [[N]] the plurality of prepare operations associated with the prepare phase to a log.

11. (Previously Presented): The method of claim 1, wherein a dedicated thread pool is used for parallel transaction operations.

12. (Previously Presented): The method of claim 1, wherein a transaction manager implements Java Transaction API.

13. (Previously Presented w): The method of claim 1, wherein the first resource is an XA resource.

14. (Previously Presented): The method of claim 6, wherein each prepare operation of N-1 of the prepare operations is associated with an XA resource.

15. (Previously Presented): The method of claim 6, wherein a dedicated thread pool is used for parallel transaction operations.

16. (Previously Presented): The method of claim 6, wherein a transaction manager implements Java Transaction API.

17. (Currently Amended): A system, comprising:

a transaction manager, wherein the transaction manager associates a plurality of resources in a transaction, wherein the plurality of resources are applied in the transaction using a prepare phase and a commit phase, wherein the prepare phase comprises a plurality of prepare operations, each of which is associated with one of the plurality of resources, and wherein the commit phase comprises a plurality of commit operations, each of which is also associated with one of the plurality of resources; and

a dedicated thread pool, on a server machine, for parallel transaction operations, including:

a first server thread, wherein the first server thread processes a first prepare operation ~~by the first server thread~~, and wherein the first prepare operation is associated with a first resource and a prepare phase;

a second server thread, wherein the first server thread dispatches a second prepare operation to the second server thread, wherein the second prepare operation is associated with a second resource and the prepare phase and processed by the second server thread in parallel to the first prepare operation being processed by the first server thread; and

a third server thread ~~a transaction manager that implements Java Transaction API~~, wherein after the transaction manager determines that the prepare phase is complete, the first server thread processes a first commit operation associated with the first resource, and the first server thread dispatches to [[a]] the third server thread a second commit operation associated with the second resource, wherein the second commit operation is processed by the third server thread in parallel to the first commit operation being processed by the first server thread.~~; and~~

~~a transaction log, wherein after the commit phase is complete, results of the commit phase are written to the transaction log~~.

18. (Previously Presented): The system of claim 17 further comprising:

a thread pool manager, wherein the thread pool manager determines available server threads.

19. (Currently Amended): The system of claim 17 further comprising:

a transaction log, wherein the transaction log records results of the prepare phase and the commit phase.

20. (Previously Presented): The method of claim 1, wherein all of the prepare operations and all of the commit operations are part of a single transaction.

21. (Previously Presented): The method of claim 6, wherein all of the prepare operations and all of the commit operations are part of a single transaction.